

## Содержание

1. Введение .....	3
2. Краткие сведения о решаемых задачах .....	4
3. Техническое задание .....	5
Требование к составу выполняемых функций .....	6
Требования к техническому обеспечению .....	6
4. Выбор средств разработки .....	7
5. Разработка приложения .....	9
Программная реализация классов .....	9
Пример работы программы .....	10
Заключение .....	14
Список литературы .....	14
ПРИЛОЖЕНИЯ .....	15

## 1. Введение

Когда существует нехватка товаров и услуг, существует неудовлетворённый спрос на необходимые товары, и уровень конкуренции будет низок, а рынок сбыта будет расти с ростом предложения товаров и услуг.

С прогрессом цивилизации у покупателя появилась возможность выбора из товара который ему нравится, так же с ростом разновидностей товаров покупатель перестает знать о наличие каких либо других товаров, ввиду того что невозможно охватить все существующие товары на рынке.

Прогресс цивилизации, для компаний, привел к тому, что предложение товара и услуги, не гарантирует сбыта продукции. Таким образом, компании столкнулись, с дефицитом потребителей. Компании стали конкурировать за деньги потребителей. Потребитель перестал реагировать на стандартную рекламу. Поэтому для активизации продажи, реклама начинает продавать товар-бренд, вместо утилитарной пользы товара говорить, о получении удовольствия от использования товара, услуги. Но даже низкая цена, реклама, маркетинг перестали гарантировать успешные продажи. Вновь появляется персонифицированные продажи.

Таким образом, на помощь компаниям пришли автоматизированные информационные системы, call-центры и аналитические БД, появилась возможность работать с каждым клиентом как с единственным. И на помощь компании приходят CRM системы. Системы, где в голову продажи ставятся взаимоотношения с клиентами.

## **2. Краткие сведения о решаемых задачах**

Система управления взаимоотношениями с клиентами (CRM, CRM-система, сокращение от англ. Customer Relationship Management) — прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности, для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путём сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов.

CRM — модель взаимодействия, полагающая, что центром всей философии бизнеса является клиент, а основными направлениями деятельности являются меры по поддержке эффективного маркетинга, продаж и обслуживания клиентов. Поддержка этих бизнес-целей включает сбор, хранение и анализ информации о потребителях, поставщиках, партнёрах, а также о внутренних процессах компании. Функции для поддержки этих бизнес-целей включают продажи, маркетинг, поддержку потребителей.

Основной целью внедрения, как правило, ставится увеличение степени удовлетворённости клиентов за счёт анализа накопленной информации о клиентском поведении, регулирования тарифной политики, настройки инструментов маркетинга. Благодаря применению автоматизированной централизованной обработки данных появляется возможность эффективно и с минимальным участием сотрудников учитывать индивидуальные потребности заказчиков, а за счёт оперативности обработки — осуществлять раннее выявление рисков и потенциальных возможностей.

Обслуживанием хранимой базы данных традиционно занимается классическая, реже распределенная СУБД, причем основной сегмент рынка в настоящий момент принадлежит решениям Oracle. Такой подход обусловлен стабильно высокой нагрузкой на базу данных, что позволяет использовать более простые способы хранения данных лишь в крайне малых организациях. Кроме того, система непрерывно обрабатывает хранимые данные, как в рамках текущей работы по обеспечению своевременной связи с контрагентами и контроля бизнес-процессов, так

и в целях определения целевых групп разного рода мероприятий, прогнозирования продаж.

В современных условиях рынка и переходе на персональную продажу товара возникает необходимость структурирования работы по продажам. На помощь в этом пришли CRM-системы. Но несмотря на большой выбор этих программ, у каждой из программ присутствуют элементы своей уникальности, которые затрудняют перемещение данных между различными системами.

### **3. Техническое задание**

Тема работы: Разработка CRM системы для работы с базой данных клиентов.

В данной программе должно быть разработано программное обеспечение, которое будет помогать пользователю взаимодействовать с базой клиентов. Приложение будет работать с БД. Так же будет реализована удобная навигация для упрощения понимания.

В соответствие с поставленной целью в работе должны быть решены следующие задачи: проектирование, разработка.

#### **Требование к составу выполняемых функций**

Разработка приложения должна обеспечивать возможность перечисленных ниже функций:

Добавление клиента;

Редактирование данных;

Удаление клиента;

Фильтр записей клиентов;

## Требования к техническому обеспечению

Приложение должно запускаться на платформе Windows. Минимальные требования представлены в таблице 1.

Таблица 1. Характеристики компьютера:

	Требования к ПК
Операционная система	WINDOWS
Процессор	Двухядерный процессор
Свободное место на диске	75 и более Мб

#### 4. Выбор средств разработки

Для разработки приложения был выбран объектно-ориентированный язык программирования C#. Язык программирования C# (произносится «си- шарп») может использоваться для разработки множества разных типов приложений. C# является основным языком разработки программ на платформе NetFramework. В нем сочетаются испытанные средства программирования с самыми последними новшествами, и предоставляется возможность для эффективного написания программ.

Мощь C# обусловлена в значительной степени поддерживаемыми им техниками, такими, например, как объектно-ориентированные возможности, обобщение и функциональное программирование. Этот язык поддерживает как динамическую, так и статическую типизацию. Благодаря поддержке технологии LINQ, он предоставляет возможности по работе со списками и множествами. А в последней версии языка также появилась встроенная поддержка асинхронного программирования.

Платформа .NET включает в себя как среду выполнения, так и библиотеки, используемые C#-программами в операционной системе Windows. В соответствии со своим названием Общезыковая среда выполнения (CLR, Common Language Runtime) поддерживает не только C#, но и любые другие языки платформы Net Framework.

Таким образом, вместе с C# мы получаем богатый набор встроенных в язык абстракций, мощную среду выполнения и лёгкий доступ к обширной библиотеке платформы.

Под платформу .NET на языке C# на данный момент существует несколько сред разработки:

- MS Visual Studio;
- MS Visual Studio Code;
- sharpDevelop;

В качестве интегрированной среды разработки было принято решение использовать Microsoft Visual Studio, как наиболее функциональную из приведённых, которая предоставляет мощный инструментарий по созданию

широкого спектра программного обеспечения. Данная среда программирования представляет собой функционально насыщенную и наиболее приспособленную под использование в организациях интегрированную среду разработки. Одним из преимуществ Visual Studio является интегрированная справочная система. Среда разработки даёт возможность получать доступ к документации непосредственно из неё, с помощью горячей клавиши F1. Интегрированный отладчик позволяет разработчику создавать нужные точки останова и отслеживать значения нужных переменных.

В редактор кода интегрировано средство IntelliSense. В системе Visual Studio название IntelliSense относится к большому количеству функций, начиная с визуальной реакции на неправильный код и интеллектуальных дескрипторов для проектирования форм до комбинаций клавиш, нажатие которых приводит к вставке целых фрагментов кода. Эти свойства, вместе взятые, обеспечивают разработчику более глубокое понимание происходящего, более высокую эффективность его работы и позволяют ему контролировать свою программу.

Для разработки программной системы было принято решение использовать бесплатную версию Visual Studio Community 2019, возможностей которой в полной мере достаточно для выполнения поставленной задачи.

## 5. Разработка приложения

### Программная реализация классов

Схема классов CustomerBase, Individual, Entity, представлена на рисунке 1.1.

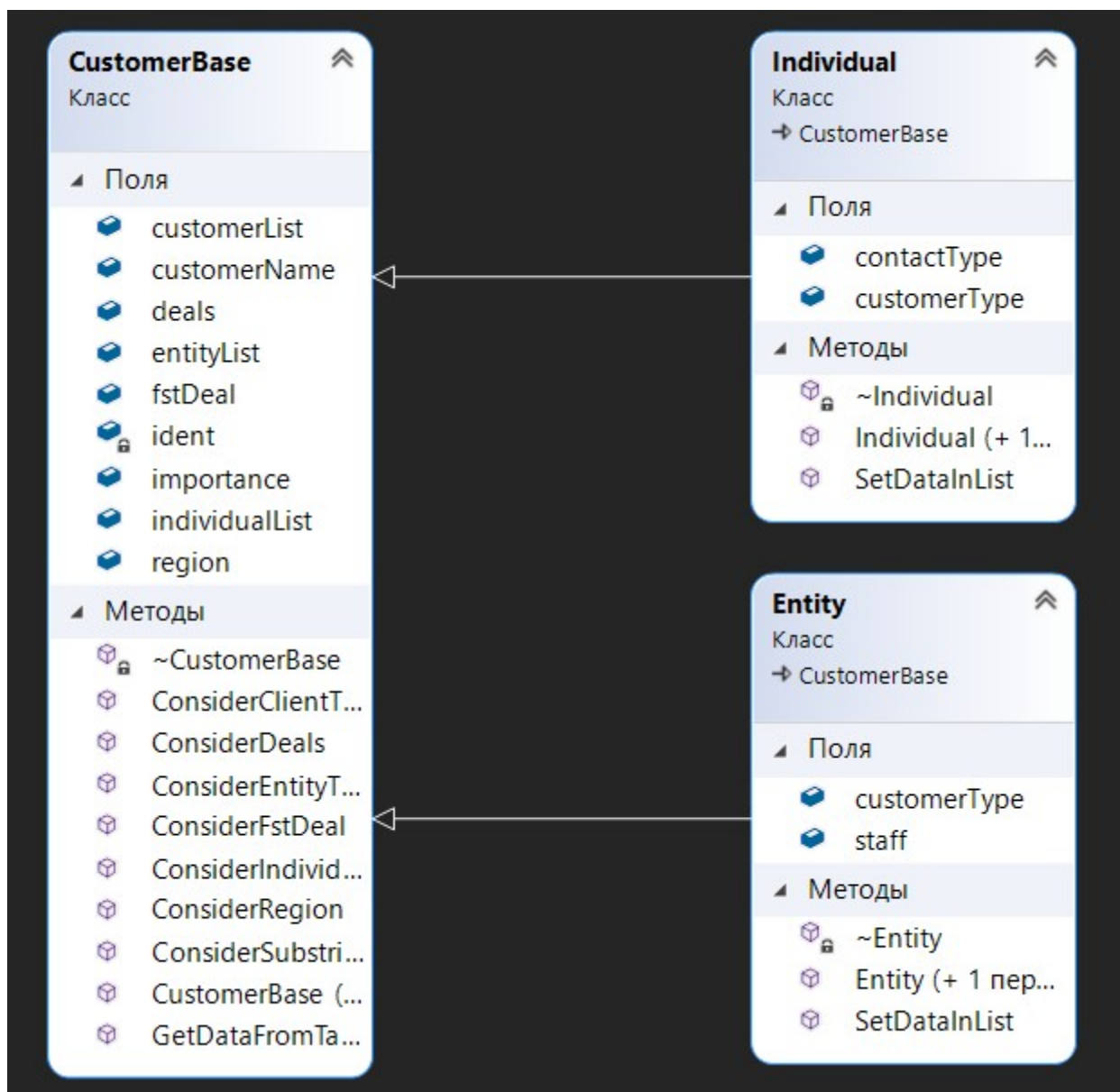


Рисунок 1.1- Схема классов



## 6. Пример работы программы

На рисунках 2.1, 2.2 и 2.3 представлено окно авторизации.

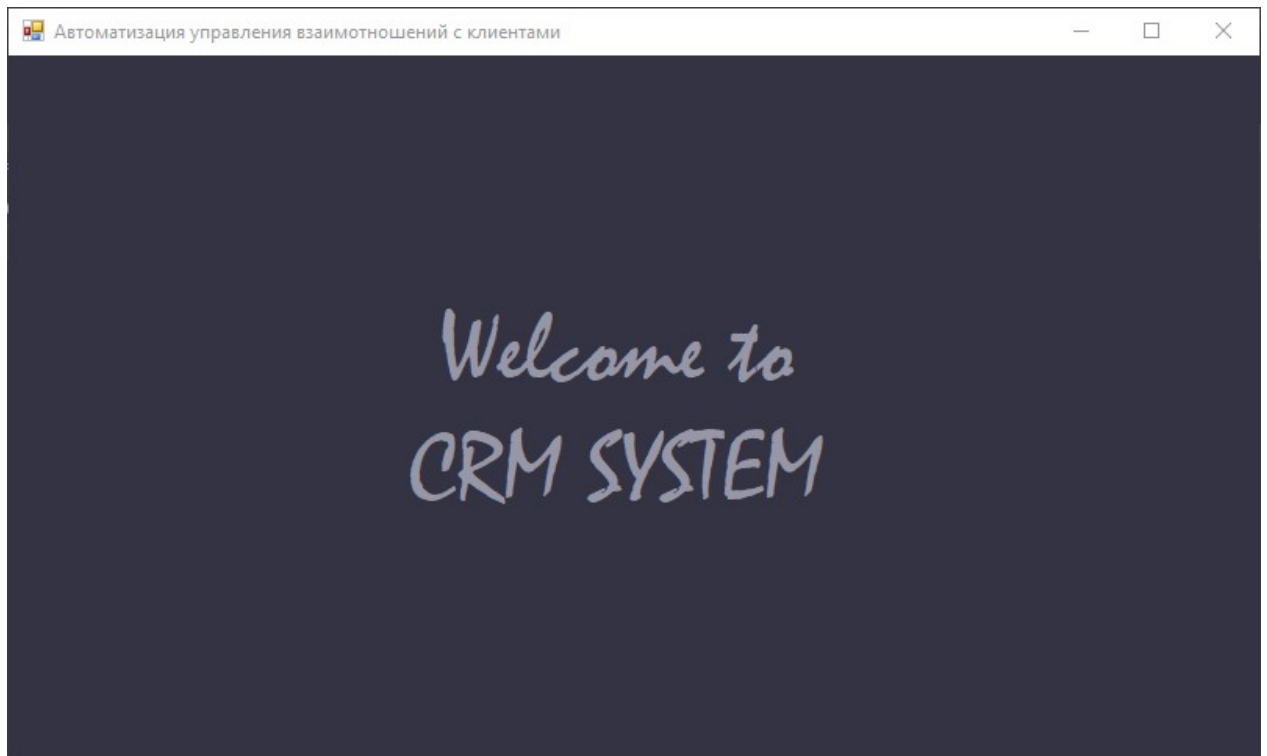


Рисунок 2.1 - Заставка

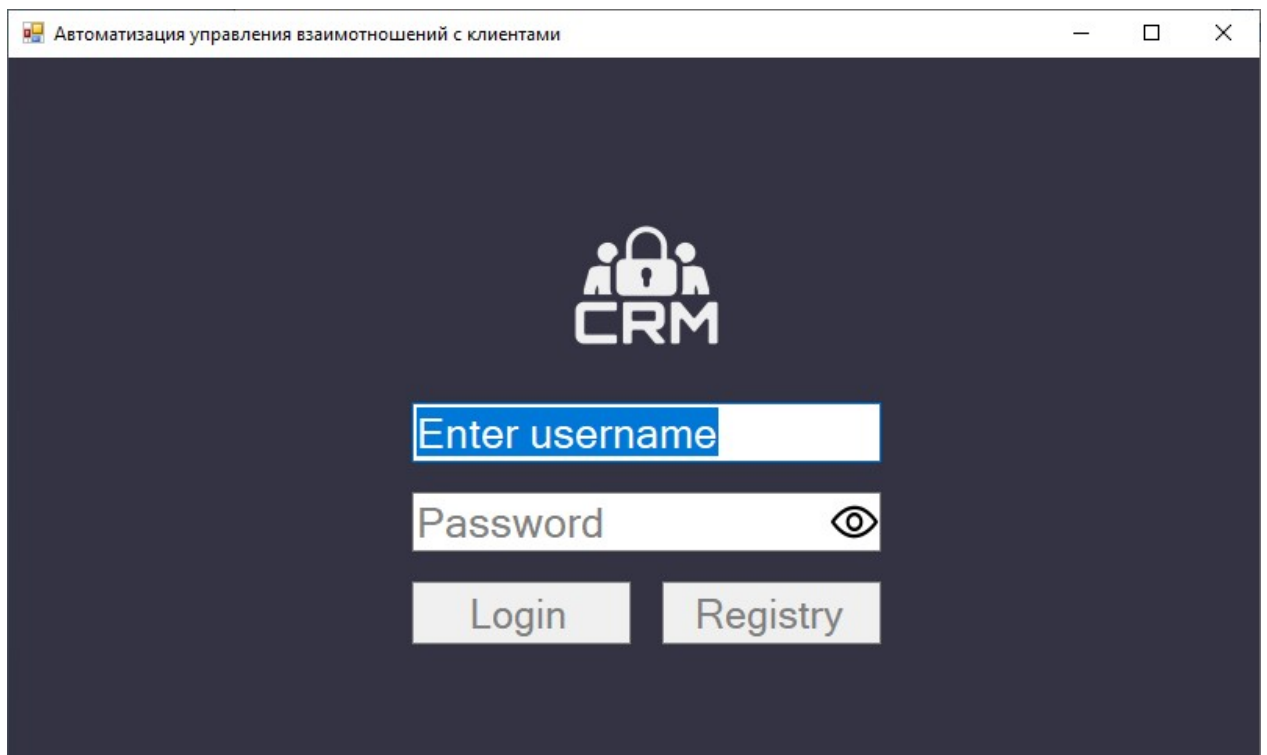


Рисунок 2.2 - Окно авторизации

Form5

CRM

Create username

Create password

Registry

Рисунок 2.3 - Окно регистрации

На рисунке 3.1 представлен интерфейс приложения.

Form1

Клиенты    Дополнительная информация    Применить фильтр    Сбросить фильтр

Наименование	Важность	Бизнес регион	Сделки	Дата первой сделки	Тип клиента
Toyota	Низкая	Япония	3	12.09.2014	Организация
Антон	Особо важный	Китай	2	05.03.2002	Физ. лицо
Microsoft	Средняя	Редмонд	1	09.08.2017	Организация
Google	Низкая	Маунтин-Вью	2	24.01.2013	Организация
Григорий	Низкая	Ижевск	2	09.02.2020	Физ. лицо
Константин	Средняя	Сарапул	5	07.03.2022	Физ. лицо
Биомир	Средняя	Ижевск	3	10.03.2022	Организация
Валерий	Средняя	Анапа	1	13.03.2022	Физ. лицо
Магнит	Средняя	Ижевск	1	16.03.2023	Организация
Золотая бухта	Особо важный	Геленджик	2	16.03.2023	Организация
Николай	Низкая	Воронеж	5	16.03.2023	Физ. лицо

Добавить нового клиента

Изменить данные выделенного клиента

Удалить выделенного клиента

Рисунок 3.1 - Интерфейс программы

Form2

Наименование

Сделки

Важность

Дата первой сделки

Бизнес регион

Тип клиента

**Добавить**

Рисунок 3.2 - Окно добавления клиента

Form2

Наименование

Сделки

Важность

Дата первой сделки

Бизнес регион

Тип клиента

Кол-во сотрудников

**Изменить**

Рисунок 3.3 - Окно изменения данных о клиенте

Form3

Наименование клиента включает в себя

Бизнес регион:

- ☐ Япония
- ☐ Китай
- ☐ Редмонд
- ☐ Маунтин-Вью
- ☐ Ижевск
- ☐ Сарапул
- ☐ Анапа
- ☐ Геленджик
- ☐ Воронеж

Диапазон количества сделок:(включительно)

От  До

Как давно в отношениях:

- ☐ Менее трех лет
- ☐ Более трех лет

☐ Физ. лицо

☐ Организация

Применить

Рисунок 3.4 - Окно фильтрации записей

## **7. Заключение**

Проведенная нами работа может показать, что боты могут упростить работу некоторых профессий.

В процессе выполнения работы была освоена технология для разработки CRM системы в приложении. Были получены навыки по созданию удобных интерфейсов на языке высокого уровня C#.

## **8. Список литературы**

1) Бьерн Страуструп. Язык программирования C# = The C# Programming Language / Пер. с англ. — 3-е изд. — СПб.; М.: Невский диалект — Бином, 1999. — 991 с.

2) Ник Рендольф, Дэвид Гарднер, Майкл Минутилло, Крис Андерсон. Visual Studio 2010 для профессионалов = Professional Visual Studio 2010. — М.: «Диалектика», 2011. — 1184 с.

3) Айвор Хортон. Microsoft Visual C# 2005: базовый курс = Beginning Visual C# 2005. — М.: «Диалектика», 2007. — 1152 с.

## 9. ПРИЛОЖЕНИЯ

### Приложение В. Программный код приложения

#### Form1.cs

```
using System;
using System.Windows.Forms;

namespace CustomerRelationshipManagement
{
    public partial class Form1 : Form
    {
        public static bool checker = false;

        public static int numbrRowsDgv;
        public Form1()
        {
            Program.f1 = this;
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            DBMS.ConnectToCustomerBase("Customers");
            Individual.SetDataInList();
            Entity.SetDataInList();
            dataGridView1.Columns[0].Visible = false;
            numbrRowsDgv = dataGridView1.Rows.Count;
        }

        private void button3_Click(object sender, EventArgs e)
        {
            Form2 addRecord = new Form2();
            Program.f2.button1.Text = "Добавить";
            addRecord.Show();
            Program.f2.textBox5.Text = (DateTime.Today.ToString()).Substring(0,10);
            Program.f2.textBox5.Enabled = false;
        }

        private void button5_Click(object sender, EventArgs e)
        {
            DBMS.DeleteSelectRecordFromBase();
        }

        private void button6_Click(object sender, EventArgs e)
        {
            if (Program.f1.dataGridView1.CurrentCell == null) MessageBox.Show("Запись не выбрана");
            else
            {
                Form2 changeRecord = new Form2();
                Program.f2.button1.Text = "Изменить";
                changeRecord.Show();
                changeRecord.LoadDataToChange();
                Program.f2.textBox5.Enabled = true;
            }
        }

        private void button8_Click(object sender, EventArgs e)
        {
            DBMS.ConnectToCustomerBase("Customers");
            checker = false;
            button3.Enabled = true;
            button5.Enabled = true;
        }
    }
}
```

```

        button6.Enabled = true;
    }

    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
    {
        if (checker == false)
        {
            int index = 0;
            for (int i = 0; i < CustomerBase.individualList.Count; i++)
            {
                if (dataGridView1.CurrentRow.Cells[1].Value.ToString() ==
CustomerBase.individualList[i].customerName) index = i;

            }

            for (int i = 0; i < CustomerBase.entityList.Count; i++)
            {
                if (dataGridView1.CurrentRow.Cells[1].Value.ToString() ==
CustomerBase.entityList[i].customerName) index = i;

            }
            switch (dataGridView1.CurrentRow.Cells[6].Value.ToString())
            {
                case «Физ. Лицо»:
                    textBox2.Text = «Полное наименование покупателя: «;
                    textBox2.Text += CustomerBase.individualList[index].customerName +
Environment.NewLine + Environment.NewLine;
                    textBox2.Text += «Важность покупателя: « +
CustomerBase.individualList[index].importance + Environment.NewLine + Environment.NewLine;
                    textBox2.Text += «Бизнес регион: « +
CustomerBase.individualList[index].region + Environment.NewLine + Environment.NewLine;
                    textBox2.Text += «Количество успешно совершенных покупок: « +
CustomerBase.individualList[index].deals + Environment.NewLine + Environment.NewLine;
                    string correctData = CustomerBase.individualList[index].fstDeal.ToString();
                    textBox2.Text += «Дата первой сделки с клиентом: « + correctData.Substring(0,
10) + Environment.NewLine + Environment.NewLine;
                    if (Int32.Parse(DateTime.Now.ToString().Substring(6, 4)) -
Int32.Parse(correctData.Substring(6, 4)) > 1)
                        textBox2.Text += «Степень доверия клиенту: « + «Высокий уровень доверия»
+ Environment.NewLine + Environment.NewLine;
                    else
                        textBox2.Text += «Степень доверия клиенту: « + «Низкий уровень доверия» +
Environment.NewLine + Environment.NewLine;
                    //textBox2.Text += CustomerBase.individualList[0].region +
Environment.NewLine;
                    textBox2.Text += «Тип покупателя: « +
CustomerBase.individualList[index].contactType + Environment.NewLine + Environment.NewLine;
                    break;
                case «Организация»:
                    textBox2.Text = «Полное наименование организации: « +
CustomerBase.entityList[index].customerName + Environment.NewLine + Environment.NewLine;
                    textBox2.Text += «Важность организации: « +
CustomerBase.entityList[index].importance + Environment.NewLine + Environment.NewLine;
                    textBox2.Text += «Бизнес регион: « + CustomerBase.entityList[index].region +
Environment.NewLine + Environment.NewLine;
                    textBox2.Text += «Количество успешно совершенных сделок: « +
CustomerBase.entityList[index].deals + Environment.NewLine + Environment.NewLine;
                    string correctDataEnt = CustomerBase.entityList[index].fstDeal.ToString();
                    textBox2.Text += «Дата начала отношений: « + correctDataEnt.Substring(0, 10)
+ Environment.NewLine + Environment.NewLine;
                    if (Int32.Parse(DateTime.Now.ToString().Substring(6, 4)) -
Int32.Parse(correctDataEnt.Substring(6, 4)) > 1)
                        textBox2.Text += «Степень доверия к организации: « + «Высокий уровень
доверия» + Environment.NewLine + Environment.NewLine;
                    else
                        textBox2.Text += «Степень доверия к организации: « + «Низкий уровень
доверия» + Environment.NewLine + Environment.NewLine;
            }
        }
    }

```

```

        textBox2.Text += "Количество сотрудников: " +
CustomerBase.entityList[index].staff + Environment.NewLine + Environment.NewLine;
        break;
    }
}
else { MessageBox.Show("Данные можно просматривать только в основной таблице"); }
}

private void button9_Click(object sender, EventArgs e)
{
    checker = true;
    DBMS.ConnectToCustomerBase("Info");
    button3.Enabled = false;
    button5.Enabled = false;
    button6.Enabled = false;
}

private void button11_Click(object sender, EventArgs e)
{
    for (int i = 0; i < numbrRowsDgv; i++) dataGridView1.Rows[i].Visible = true;
    button11.Enabled = false;
    button10.Enabled = true;
}

private void button10_Click(object sender, EventArgs e)
{
    Form3 filter = new Form3();
    filter.Show();
}

```

## Form5.cs

```

using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CustomerRelationshipManagement
{
    public partial class Form4 : Form
    {
        public static byte r_b = 51, g_b = 51, b_b = 67;           //background color
        public static byte r_f = 240, g_f = 240, b_f = 240;       //font color
        public byte r = r_b, g = g_b, b = b_b;
        public bool eyeHided = true;
        public TextBox pubBox;
        public TextBox pubBoxL;
        public Label globlab;

        private void button1_Click(object sender, EventArgs e)
        {

        }

        public Form4()
        {
            Program.f4 = this;
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private async void Form4_Load(object sender, EventArgs e)

```



```

{
    //timer1.Enabled = true;
    while (b <= b_f)
    {
        r += 2;
        g += 2;
        b += 2;
        label1.ForeColor = Color.FromArgb(r, g, b);
        await Task.Delay(5);
    }
    while (b > b_b)
    {
        r -= 2;
        g -= 2;
        b -= 2;
        label1.ForeColor = Color.FromArgb(r, g, b);
        await Task.Delay(10);
    }
    this.Controls.Clear();
    TextBox loginBox = new TextBox();
    TextBox passwordBox = new TextBox();
    PictureBox pctBox = new PictureBox();
    PictureBox userImg = new PictureBox();
    PictureBox passwordImg = new PictureBox();
    PictureBox eyePass = new PictureBox();
    Button btnLog = new Button();
    Button btnReg = new Button();
    Label lblError = new Label();
    pctBox.Image = CustomerRelationshipManagement.Properties.Resources.loginicon;
    pctBox.SizeMode = PictureBoxSizeMode.StretchImage;
    pctBox.Size = new Size(200, 200);
    userImg.Image = CustomerRelationshipManagement.Properties.Resources.usericon;
    userImg.SizeMode = PictureBoxSizeMode.StretchImage;
    userImg.Size = new Size(40, 40);
    passwordImg.Image = CustomerRelationshipManagement.Properties.Resources.passwordi;
    passwordImg.SizeMode = PictureBoxSizeMode.StretchImage;
    passwordImg.Size = new Size(40, 40);
    eyePass.Image = CustomerRelationshipManagement.Properties.Resources.eye;
    eyeHided = false;
    eyePass.SizeMode = PictureBoxSizeMode.StretchImage;
    eyePass.Size = new Size((int)(passwordBox.Height * 1.5), passwordBox.Height);
    eyePass.Dock = DockStyle.Right;
    loginBox.Size = new Size(300, 40);
    passwordBox.Size = new Size(300, 40);
    btnLog.Size = new Size(140, 40);
    btnReg.Size = new Size(140, 40);
    loginBox.Font = new Font(loginBox.Font.FontFamily, loginBox.Size.Height);
    passwordBox.Font = new Font(passwordBox.Font.FontFamily, passwordBox.Size.Height);
    lblError.AutoSize = true;
    //loginBox.BorderStyle = BorderStyle.Fixed3D;
    loginBox.Text = "Enter username";
    passwordBox.Text = "Password";
    loginBox.ForeColor = Color.Gray;
    passwordBox.ForeColor = Color.Gray;
    btnLog.BackColor = SystemColors.Control;
    btnLog.FlatStyle = FlatStyle.Flat;
    btnLog.ForeColor = Color.Gray;
    btnLog.Font = new Font(btnLog.Font.FontFamily, 19);
    btnLog.Text = "Login";
    btnReg.BackColor = SystemColors.Control;
    btnReg.FlatStyle = FlatStyle.Flat;
    btnReg.ForeColor = Color.Gray;
    btnReg.Font = new Font(btnLog.Font.FontFamily, 19);
    btnReg.Text = "Registry";
    loginBox.Click += new EventHandler(btnLoginClick);
    passwordBox.Click += new EventHandler(btnPasswordClick);
    loginBox.TextChanged += new EventHandler(txtLogChanging);
    eyePass.Click += new EventHandler(clickEye);
    eyePass.Move += new EventHandler(onEye);
}

```

```

        btnReg.Click += new EventHandler(AddNewUser);
        btnLog.Click += new EventHandler(CheckUser_Auth);
        loginBox.Location = new Point(this.Width/2 - loginBox.Width/2, 220);
        passwordBox.Location = new Point(this.Width / 2 - passwordBox.Width / 2,
loginBox.Location.Y + (int)(loginBox.Height * 1.5));
        btnLog.Location = new Point(/*this.Width / 2 - btnLog.Width / 2*/passwordBox.Location.X,
loginBox.Location.Y + (int)(loginBox.Height * 3));
        btnReg.Location = new Point(/*this.Width / 2 - btnLog.Width / 2*/passwordBox.Location.X +
160, loginBox.Location.Y + (int)(loginBox.Height * 3));
        pctBox.Location = new Point(this.Width / 2 - pctBox.Width / 2, loginBox.Location.Y -
175);

        userImg.Location = new Point(loginBox.Location.X - 50, loginBox.Location.Y);
        passwordImg.Location = new Point(passwordBox.Location.X - 50, passwordBox.Location.Y);
        lblError.Location = new Point(loginBox.Location.X, loginBox.Location.Y + 98);
        this.Controls.Add(loginBox);
        this.Controls.Add(passwordBox);
        this.Controls.Add(pctBox);
        this.Controls.Add(btnLog);
        this.Controls.Add(btnReg);
        this.Controls.Add(lblError);
        passwordBox.Controls.Add(eyePass);
        pubBox = passwordBox;
        pubBoxL = loginBox;
        globlab = lblError;
    }

    public void clickEye(object sender, EventArgs eventHandler) {
        PictureBox pBox = sender as PictureBox;
        if (eyeHided)
        {
            pBox.Image = Properties.Resources.eye;
            eyeHided = false;
            pubBox.UseSystemPasswordChar = false;
        }
        else
        {
            pBox.Image = Properties.Resources.hide;
            eyeHided = true;
            pubBox.UseSystemPasswordChar = true;
            if (pubBox.Text == "Password") pubBox.Text = "";
        }
    }

    public void onEye(object sender, EventArgs eventHandler)
    {
        PictureBox pBox = sender as PictureBox;
        pBox.Cursor = Cursors.Hand;
    }

    public void btnLoginClick(object sender, EventArgs eventHandler) {
        TextBox textBox = sender as TextBox;
        textBox.SelectionStart = 0;
        textBox.Text = "";
    }
    public void btnPasswordClick(object sender, EventArgs eventHandler)
    {
        TextBox textBox = sender as TextBox;
        textBox.SelectionStart = 0;
        textBox.Text = "";
        pubBox.UseSystemPasswordChar = true;
    }
    public void txtLogChanging(object sender, EventArgs eventHandler)
    {
        TextBox txtBox = sender as TextBox;

```

```

    }
    public void AddNewUser(object sender, EventArgs eventHandler)
    {
        Form5 frm = new Form5();
        frm.Show();
    }

    public async void CheckUser_Auth(object sender, EventArgs eventHandler)
    {
        System.IO.StreamReader authCheck = new System.IO.StreamReader("Users.txt");
        string lineLog = authCheck.ReadLine();
        if (lineLog.Substring(0, lineLog.IndexOf(":")) == pubBoxL.Text
            && lineLog.Substring(lineLog.IndexOf(":") + 1, lineLog.Length - lineLog.IndexOf(":")
- 1) == pubBox.Text)
        {
            globlab.Text = "Login successfull";
            globlab.ForeColor = Color.GreenYellow;
            authCheck.Close();
            await Task.Delay(1000);
            Form1 f1 = new Form1();
            f1.Show();
            this.Visible = false;
        }
        else
        {
            globlab.Text = "Invalid user";
            globlab.ForeColor = Color.Red;
            authCheck.Close();
        }
    }
}

```

## DBMS.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace CustomerRelationshipManagement
{
    class DBMS
    {
        public static SqlConnection sqlConnection = null;
        public static SqlCommandBuilder sqlBuilder = null;
        public static SqlDataAdapter sqlDataAdapter = null;
        public static DataSet dataSet = null;
        public const string connectionString = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\PC\Desktop\Летняя практика
#3\last\CustomerRelationshipManagement\CustomerRelationshipManagement\Database1.mdf;Integrated
Security=True";

        public static void ConnectToCustomerBase(string baseName) {
            sqlConnection = new SqlConnection(connectionString);
            sqlConnection.Open();
            try
            {
                sqlDataAdapter = new SqlDataAdapter($"SELECT * FROM {baseName}", sqlConnection);
                sqlBuilder = new SqlCommandBuilder(sqlDataAdapter);
                sqlBuilder.GetInsertCommand();
                sqlBuilder.GetUpdateCommand();
                sqlBuilder.GetDeleteCommand();
                dataSet = new DataSet();
                sqlDataAdapter.Fill(dataSet, $"{baseName}");
                Program.f1.dataGridView1.DataSource = dataSet.Tables[$"{baseName}"];
            }
        }
    }
}

```

```

        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    public static void ConnectToCustomerInfoBase()
    {
        sqlConnection = new SqlConnection(connectionString);
        sqlConnection.Open();
        try
        {
            sqlDataAdapter = new SqlDataAdapter($"SELECT * FROM Info", sqlConnection);
            sqlBuilder = new SqlCommandBuilder(sqlDataAdapter);
            sqlBuilder.GetInsertCommand();
            sqlBuilder.GetUpdateCommand();
            sqlBuilder.GetDeleteCommand();
            dataSet = new DataSet();
            sqlDataAdapter.Fill(dataSet, "Info");
            Program.f1.dataGridView1.DataSource = dataSet.Tables["Info"];
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    public static async void AddRecordToCustomerBase() {
        sqlConnection = new SqlConnection(connectionString);
        sqlConnection.Open();
        int a;
        if (!string.IsNullOrEmpty(Program.f2.textBox1.Text) &&
            !string.IsNullOrEmpty(Program.f2.textBox2.Text)
            && !string.IsNullOrEmpty(Program.f2.textBox3.Text) &&
            !string.IsNullOrEmpty(Program.f2.textBox4.Text)
            && !string.IsNullOrEmpty(Program.f2.textBox5.Text) &&
            int.TryParse(Program.f2.textBox4.Text, out a))
        {
            string cmdStr;
            //if (mode == false)
            //{
                cmdStr = $"INSERT INTO [Customers] ([{Program.f2.label1.Text}],
                [{Program.f2.label2.Text}], [{Program.f2.label3.Text}], [{Program.f2.label4.Text}],
                [{Program.f2.label5.Text}], [{Program.f2.label6.Text}])VALUES(@v1, @v2, @v3, @v4, @v5, @v6)";
            //}
            //else
            //{
                // cmdStr = $"UPDATE [{Program.f1.dataGridView1.DataSource}] SET [{label1.Text}] =
                @v1, [{label2.Text}] = @v2, [{label3.Text}] = @v3, [Занято ячеек] = @v5, Производитель = @v6 Where" +
                $" [{Program.f1.dataGridView1.Columns[1].HeaderText}] = @v1 ";
            //}
            SqlCommand command = new SqlCommand(cmdStr, sqlConnection);
            command.Parameters.AddWithValue("v1", Program.f2.textBox1.Text);
            command.Parameters.AddWithValue("v2", Program.f2.textBox2.Text);
            command.Parameters.AddWithValue("v3", Program.f2.textBox3.Text);
            //command.Parameters.AddWithValue("v4", Program.f2.currentRack);
            command.Parameters.AddWithValue("v4", Program.f2.textBox4.Text);
            command.Parameters.AddWithValue("v5", DateTime.Today);
            command.Parameters.AddWithValue("v6", Program.f2.comboBox1.SelectedItem);
            await command.ExecuteNonQueryAsync();
            sqlConnection.Close();
            //Program.f1.button2.Enabled = true;
            Program.f2.res = true;

            Program.f2.Close();
        }
        else
    }

```

```

        {
            MessageBox.Show("Не все данные заполнены верно.");
            Program.f2.res = false;
        }
    }

    public static async void DeleteSelectRecordFromBase() {
        SqlConnection sqlConnection = null;
        sqlConnection = new SqlConnection(connectionString);
        sqlConnection.Open();
        DialogResult dlgRes = MessageBox.Show("Вы уверены, что хотите удалить данную строку?",
"Удаление записи", MessageBoxButtons.YesNo,
            MessageBoxIcon.Question);
        if (dlgRes == DialogResult.Yes)
        {
            SqlCommand command = new SqlCommand($"DELETE FROM
[{Program.f1.dataGridView1.DataSource}] WHERE [{Program.f1.dataGridView1.Columns[1].HeaderText}] =
@name", sqlConnection);
            command.Parameters.AddWithValue("name",
Program.f1.dataGridView1.CurrentRow.Cells[1].Value);
            await command.ExecuteNonQueryAsync();
            sqlConnection.Close();
            //Program.f1.button2.Enabled = true;
        }
        else if (dlgRes == DialogResult.No)
        {
        }
        else
        {
            MessageBox.Show("Вы заполнили не все данные либо пытаетесь удалить неверную
запись.");
        }
    }

}

static public async void ChangeRecordDataInBase()
{
    MessageBox.Show("Изменения работают");

    SqlConnection sqlConnection = null;
    sqlConnection = new SqlConnection(connectionString);
    sqlConnection.Open();
    int a;
    string dateCheck = Program.f2.textBox5.Text + " 0:00:00";
    string dtachck = "";
    string dtachck1 = "";
    DateTime temp;
    if (int.TryParse(dateCheck.Substring(0, 2), out a) && int.TryParse(dateCheck.Substring(3,
2), out a))
    {
        dtachck = dateCheck.Substring(0, 2);
        dtachck1 = dateCheck.Substring(3, 2);
    }
    else
    {
        dtachck = "32";
        dtachck1 = "13";
    }
    if (dateCheck.Contains(',') || !(DateTime.TryParse(dateCheck, out temp)) || dateCheck ==
"" || int.Parse(dtachck) > 31 || int.Parse(dtachck1) > 12)
    {
        MessageBox.Show("Неверно заполнено поле <Дата рождения>. \nИспользуйте формат
<ДД.ММ.ГГГГ>." + DateTime.TryParse(dateCheck, out temp), "Неверный ввод данных",
MessageButtons.OK, MessageBoxIcon.Error);
    }
    else if (!string.IsNullOrEmpty(Program.f2.textBox1.Text) &&
!string.IsNullOrEmpty(Program.f2.textBox2.Text)

```

```

        && !string.IsNullOrEmpty(Program.f2.textBox3.Text) &&
!string.IsNullOrEmpty(Program.f2.textBox4.Text)
        && !string.IsNullOrEmpty(Program.f2.textBox5.Text) &&
(Program.f2.comboBox1.Text.ToString()=="Физ. лицо" || Program.f2.comboBox1.Text.ToString() ==
"Организация"))
    {
        string cmdStr = $"UPDATE [Customers] SET [{Program.f2.label1.Text}] = @v1, " +
            $"[{Program.f2.label2.Text}] = @v2, [{Program.f2.label3.Text}] = @v3,
[{Program.f2.label4.Text}] = @v4," +
            $" [{Program.f2.label5.Text}] = @v5, [{Program.f2.label6.Text}] = @v6 Where" +
            $" [{Program.f1.dataGridView1.Columns[1].HeaderText}] = @v7 ";
        SqlCommand command = new SqlCommand(cmdStr, sqlConnection);
        command.Parameters.AddWithValue("v1", Program.f2.textBox1.Text);
        command.Parameters.AddWithValue("v2", Program.f2.textBox2.Text);
        command.Parameters.AddWithValue("v3", Program.f2.textBox3.Text);
        //command.Parameters.AddWithValue("v4", Program.f2.currentRack);
        command.Parameters.AddWithValue("v4", Program.f2.textBox4.Text);
        command.Parameters.AddWithValue("v5", DateTime.Parse(dateCheck));
        command.Parameters.AddWithValue("v6", Program.f2.comboBox1.SelectedItem);
        command.Parameters.AddWithValue("v7",
Program.f1.dataGridView1.CurrentRow.Cells[1].Value.ToString());

        await command.ExecuteNonQueryAsync();
        sqlConnection.Close();
        // Program.f1.button2.Enabled = true;
        Program.f2.res = true;
        Program.f2.Close();
    }
    else
    {
        MessageBox.Show("Не все данные были заполнены верно.");
        Program.f2.res = false;
    }
}

public static async void AddRecordToInfoBase()
{
    sqlConnection = new SqlConnection(connectionString);
    sqlConnection.Open();
    int a;
    if (!string.IsNullOrEmpty(Program.f2.textBox1.Text) &&
!string.IsNullOrEmpty(Program.f2.textBox7.Text))
    {
        string cmdStr;
        //if (mode == false)
        //{
        cmdStr = $"INSERT INTO [Info] ([Информация], [Клиент])VALUES(@v1, @v2)";
        //}
        //else
        //{
        //    cmdStr = $"UPDATE [{Program.f1.dataGridView1.DataSource}] SET [{label1.Text}] =
@v1, [{label2.Text}] = @v2, [{label3.Text}] = @v3, [Занято ячеек] = @v5, Производитель = @v6 Where" +
        //    $" [{Program.f1.dataGridView1.Columns[1].HeaderText}] = @v1 ";
        //}
        SqlCommand command = new SqlCommand(cmdStr, sqlConnection);
        command.Parameters.AddWithValue("v1", Program.f2.textBox7.Text);
        command.Parameters.AddWithValue("v2", Program.f2.textBox1.Text);
        await command.ExecuteNonQueryAsync();
        sqlConnection.Close();
        //Program.f1.button2.Enabled = true;
        //Program.f2.res = true;

        Program.f2.Close();
    }
    else
    {
        MessageBox.Show("Не все данные заполнены верно.");
    }
}

```

```

        ///Program.f2.res = false;
    }
}

public static async void ChangeRecordInfoBase() {
    sqlConnection = new SqlConnection(connectionString);
    sqlConnection.Open();
    int a;

    if (!string.IsNullOrEmpty(Program.f2.textBox1.Text) &&
        !string.IsNullOrEmpty(Program.f2.textBox7.Text))
    {
        string cmdStr;
        //if (mode == false)
        //{
            cmdStr = $"UPDATE [Info] SET [Информация] = @v1, [Клиент] = @v2 Where [Клиент] =
@v3";
            //}
            //else
            //{
                cmdStr = $"UPDATE [{Program.f1.dataGridView1.DataSource}] SET [{label1.Text}] =
@v1, [{label2.Text}] = @v2, [{label3.Text}] = @v3, [Занято ячеек] = @v5, Производитель = @v6 Where " +
                $" [{Program.f1.dataGridView1.Columns[1].HeaderText}] = @v1 ";
            //}
            SqlCommand command = new SqlCommand(cmdStr, sqlConnection);
            command.Parameters.AddWithValue("v1", Program.f2.textBox7.Text);
            command.Parameters.AddWithValue("v2", Program.f2.textBox1.Text);
            command.Parameters.AddWithValue("v3",
Program.f1.dataGridView1.CurrentRow.Cells[1].Value.ToString());
            await command.ExecuteNonQueryAsync();
            sqlConnection.Close();
            //Program.f1.button2.Enabled = true;
            //Program.f2.res = true;
            //ConnectToCustomerBase("Customers");
            Program.f2.Close();
        }
        else
        {
            MessageBox.Show("Не все данные заполнены верно.");
            /// Program.f2.res = false;
        }
    }
}
}
}

```